

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-187558

(43) 公開日 平成10年(1998) 7月21日

(51) Int.Cl. <sup>8</sup>	識別記号	F I
G 0 6 F 13/00	3 5 1	G 0 6 F 13/00
9/46	3 6 0	9/46
15/16	4 6 0	15/16
		3 5 1 C
		3 6 0 F
		4 6 0 T

審査請求 未請求 請求項の数 7 O L (全 13 頁)

(21) 出願番号 特願平8-339717

(22) 出願日 平成8年(1996)12月19日

(71) 出願人 000001443

カシオ計算機株式会社

東京都渋谷区本町1丁目6番2号

(72) 発明者 矢代 義徳

東京都羽村市栄町3丁目2番1号 カシオ  
計算機株式会社羽村技術センター内

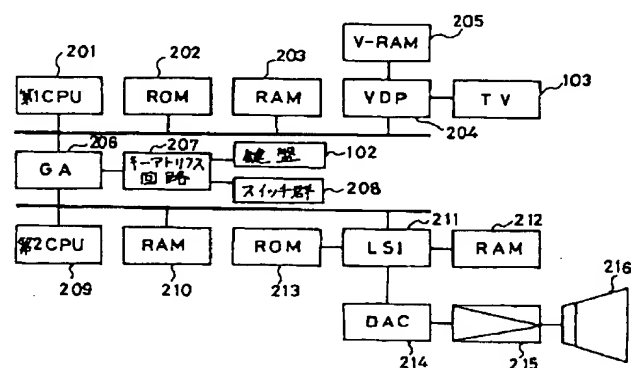
(74) 代理人 弁理士 阪本 紀康

(54) 【発明の名称】 データ転送方法、及びデータ処理装置

(57) 【要約】

【課題】 マルチプロセッサ方式を採用した装置等において、データ転送に要する時間を短縮して処理の効率を向上させる。

【解決手段】 第1CPU201は、ROM202内のプログラムを実行することにより、スイッチ処理、鍵盤処理といった主要なサブルーチン処理、それらのサブルーチン処理を実行することで生成した発音制御データを第2CPU209に転送する処理を行う。そのデータを転送する処理では、ROM202に格納された楽音制御データ処理時間データ群や主要ルーチン実行時間データ群等から、再び転送処理を実行するまでに処理が完了しないだけの発音制御データが第2CPU209に転送されていると予測した場合、第2CPU209に転送すべきデータの転送を行わずに他のサブルーチン処理に移行する。



第1の実施の形態に適用されたシステムの構成図

1

## 【特許請求の範囲】

【請求項 1】 第 1 の処理手段が第 2 の処理手段に、該第 2 の処理手段に処理させるデータを転送する方法であって、

前記第 2 の処理手段が処理に必要とする処理時間をデータの種類別に予め用意し、

前記第 2 の処理手段に転送すべきデータが生じた場合に、前記処理時間から前記第 2 の処理手段が転送済みのデータに対する処理を完了するまでの処理残り時間を予測し、

該予測した処理残り時間に基づいて、前記転送すべきデータを前記第 1 の処理手段から前記第 2 の処理手段に転送させる、

ことを特徴とするデータ転送方法。

【請求項 2】 第 1 の処理手段が第 2 の処理手段に、該第 2 の処理手段に処理させるデータを転送する方法であって、

前記第 2 の処理手段が処理に必要とする処理時間をデータの種類別に予め用意し、

前記第 1 の処理手段がルーチン処理の実行に必要とする実行時間をルーチン処理の種類別に用意し、

前記第 1 の処理手段が前記第 2 の処理手段にデータ転送を行う転送処理を前記ルーチン処理として実行したとき、前記第 2 の処理手段に転送すべきデータが生じていた場合に、前記処理時間から前記第 2 の処理手段が転送済みのデータの処理を完了するまでの処理残り時間、及び、前記実行時間を参照して、前記転送処理を再び実行するまでの実行間隔をそれぞれ予測し、

前記処理残り時間、及び実行間隔の各予測結果に基づいて、前記転送すべきデータを前記第 1 の処理手段から前記第 2 の処理手段に転送させる、

ことを特徴とするデータ転送方法。

【請求項 3】 前記実行時間は、前記第 1 の処理手段が各ルーチン処理の実行に要した実際の実行時間を計測して取得する、

ことを特徴とする請求項 2 記載のデータ転送方法。

【請求項 4】 所定の第 1 の処理、及び該第 1 の処理の実行により得た転送すべきデータを転送する転送処理をルーチン処理として少なくとも実行する第 1 の処理手段と、該第 1 の処理手段から転送されたデータを用いて処理を実行する第 2 の処理手段とを備えたデータ処理装置において、

前記第 1 の処理手段が前記転送処理を実行したとき、前記第 2 の処理手段が転送済みのデータの処理を完了するのに必要とする処理残り時間を予測する第 1 の予測手段と、

前記第 1 の予測手段が予測した処理残り時間に基づいて、前記第 1 の処理手段による前記第 2 の処理手段へのデータ転送を制御する制御手段と、

を具備したことを特徴とするデータ処理装置。

2

【請求項 5】 前記第 1 の処理手段が前記転送処理を実行してから次にそれを実行するまでの実行間隔を予測する第 2 の予測手段を更に具備し、

前記制御手段は、前記第 1 及び第 2 の予測手段の各予測結果に基づいて、前記第 1 の処理手段による前記第 2 の処理手段へのデータ転送を制御する、

ことを特徴とする請求項 4 記載のデータ処理装置。

【請求項 6】 前記第 2 の予測手段は、前記第 1 の処理手段が実行した前記転送処理の実際の実行間隔を計測して取得し、該取得した過去の実行間隔に基づいて、前記実行間隔を予測する、

ことを特徴とする請求項 5 記載のデータ処理装置。

【請求項 7】 前記制御手段は、前記第 1 及び第 2 の予測手段の少なくとも一つの予測結果に基づいて転送させるべきデータの種類の選択し、該選択した種類のデータを前記転送すべきデータのなかから抽出して前記第 1 の処理手段に転送させる、

ことを特徴とする請求項 4、5、または 6 記載のデータ処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、転送先の負荷の重さを予測してデータを転送する技術に関する。

【0002】

【従来の技術および発明が解決しようとする課題】例えば電子楽器では、近年多機能化が進み、数多くの機能が搭載される傾向にある。楽音を発音するという本来の機能の他に、音符や絵を表示させる画像表示機能等が搭載されることも多くなっている。

【0003】搭載される機能が多くなるほど、全体の処理の負荷は大きくなる。搭載される機能を実現させるための処理を 1 つの CPU に集中させると、一通りの処理を実行するのに要する時間が長くなり、動作が遅くなるという不具合が発生する。そのため、電子楽器においても、搭載された機能を実現するために、マルチプロセッサ方式が採用されることが多くなっている。

【0004】そのマルチプロセッサ方式では、ある CPU（以降、他との区別のために第 1 CPU と呼ぶ）が他の CPU（以降、他との区別のために第 2 CPU と呼ぶ）にデータを転送し、処理を依頼する形になる。従来のデータ転送方法では、第 1 CPU から第 2 CPU へのデータ転送を、第 2 CPU 側に用意されているデータ格納領域（バッファ）の空容量を確認して行われていた。具体的には、例えば第 2 CPU にデータ転送を行う場合、第 1 CPU は第 2 CPU にデータ格納領域の空容量を問い合わせ、転送すべきデータを格納できるだけの容量を第 2 CPU が用意したのを確認した後、データ転送を行っていた。

【0005】しかし、上記のデータ転送方法では、空容量の有無を確認する期間や第 2 CPU が空容量の確保を

3

行っている期間、第1CPUは実質的に待ち状態となる。そのため、データ転送に長い時間がかかり、それによってスループットが低下することから、処理を効率的に行えないという問題点があった。

【0006】本発明の課題は、データ転送に要する時間を短縮して処理の効率を向上させることにある。

【0007】

【課題を解決するための手段】本発明の第1の態様のデータ転送方法は、第1の処理手段が第2の処理手段に、該第2の処理手段に処理させるデータを転送するためのものであり、第2の処理手段が処理に必要とする処理時間をデータの種類の別に予め用意し、第2の処理手段に転送すべきデータが生じた場合に、処理時間から第2の処理手段が転送済みのデータに対する処理を完了するまでの処理残り時間を予測し、該予測した処理残り時間に基づいて、転送すべきデータを第1の処理手段から第2の処理手段に転送させる。

【0008】本発明の第2の態様のデータ転送方法は、第1の処理手段が第2の処理手段に、該第2の処理手段に処理させるデータを転送するためのものであり、第2の処理手段が処理に必要とする処理時間をデータの種類の別に予め用意し、第1の処理手段がルーチン処理の実行に必要とする実行時間をルーチン処理の種類の別に用意し、第1の処理手段が第2の処理手段にデータ転送を行う転送処理をルーチン処理として実行したとき、第2の処理手段に転送すべきデータが生じていた場合に、処理時間から第2の処理手段が転送済みのデータの処理を完了するまでの処理残り時間、及び、実行時間を参照して、転送処理を再び実行するまでの実行間隔をそれぞれ予測し、処理残り時間、及び実行間隔の各予測結果に基づいて、転送すべきデータを第1の処理手段から第2の処理手段に転送させる。

【0009】なお、上記実行時間は、第1の処理手段が各ルーチン処理の実行に要した実際の実行時間を計測して取得する、ことが望ましい。本発明の第1の態様のデータ処理装置は、所定の第1の処理、及び該第1の処理の実行により得た転送すべきデータを転送する転送処理をルーチン処理として少なくとも実行する第1の処理手段と、該第1の処理手段から転送されたデータを用いて処理を実行する第2の処理手段とを備えていることを前提として、第1の処理手段が転送処理を実行したとき、第2の処理手段が転送済みのデータに対する処理に必要とする処理残り時間を予測する第1の予測手段と、第1の予測手段が予測した処理残り時間に基づいて、第1の処理手段による第2の処理手段へのデータ転送を制御する制御手段と、を具備する。

【0010】本発明の第2の態様のデータ処理装置は、上記第1の態様の構成に加えて、第1の処理手段が転送処理を実行してから次にそれを実行するまでの実行間隔を予測する第2の予測手段を更に具備し、制御手段は、

4

第1及び第2の予測手段の各予測結果に基づいて、第1の処理手段による第2の処理手段へのデータ転送を制御する。

【0011】なお、上記の構成において、第2の予測手段は、第1の処理手段が実行した転送処理の実際の実行間隔を計測して取得し、該取得した過去の実行間隔に基づいて、実行間隔を予測する、ことが望ましい。また、制御手段は、第1及び第2の予測手段の少なくとも一つの予測結果に基づいて転送させるべきデータの種類の選択し、該選択した種類のデータを転送すべきデータのなかから抽出して第1の処理手段に転送させる、ことが望ましい。

【0012】本発明では、転送元がデータの転送先にデータを転送する場合、転送先の処理能力、転送済みのデータ量、その種類等から転送先の負荷の状態を予測し、その予測結果に基づいてデータを転送する。これにより、転送元は、転送先にその負荷の状態や空容量についての問い合わせを行うことなく、負荷の状態や空容量の状態といった転送先の状態に合わせたデータ転送や、更に割り当てられている処理といった自身の状態も考慮したデータ転送が可能となる。

【0013】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態につき詳細に説明する。

<第1の実施の形態>図1は、第1の実施の形態が適用されたシステムの概観図である。

【0014】図1に示すように、上記システムは、ユーザに演奏操作される鍵盤102を備えた電子楽器101と、それにケーブル104によって接続されたテレビ103とから構成されている。第1の実施の形態によるデータ処理装置は、電子楽器101に搭載されることでシステムに適用されている。

【0015】図2は、図1に示すシステムの構成図である。次に、図2を参照して、その構成、及び動作について説明する。第1CPU201は、ROM202に格納されているプログラムを読み出して実行することにより、RAM203をワーク領域として使用しながら楽器101全体を制御する。その制御を通して、テレビ103にユーザが所望する楽譜等の画像を表示させる。

【0016】上記ROM202には、第1CPU201が実行するプログラムの他に、自動演奏用の演奏データや各種制御用データ、更にはテレビ103に画像を表示させるための画像データが格納されている。その画像データは、例えば画像を構成する部分画像（スプライト）毎に用意されている。

【0017】第1CPU201は、画像データを表示させる場合、ROM202から画像データを読み出してそれをRAM203に格納した後、VDP（ビデオ・ディスプレイ・プロセッサ）204に転送する。また、VDP204に、各画像データを表示させる座標データも合

10

20

30

40

50

5

わせて転送する。

【0018】他方のVDP204は、第1CPU201から転送された画像データをV-RAM(ビデオRAM)205に格納し、座標データを内部のメモリに格納する。そのV-RAM205に格納した画像データは、座標データに従って読み出し、読み出した画像データを予め定められた優先順位に従って合成(重ね合わせ)する。その合成が行われた画像データは、RGBデータとしてVDP204から出力される。そのRGBデータは、不図示のエンコーダによってNTSC方式の映像信号に変換された後、端子、及び図1に示すケーブル104を介してテレビ103に出力される。それにより、テレビ103の画面に例えば図1に示すような楽譜が表示される。

【0019】第1CPU201は、鍵盤102に対してユーザが行った操作を、ゲートアレイ(GA)206、及びキーマトリクス回路207を介して検出する。より具体的には、キーマトリクス回路207を用いて鍵盤102をスキャンし、ゲートアレイ206を介してスキャン結果を受け取り、その受け取ったスキャン結果を解析することにより、鍵盤102に対してユーザが行った操作を検出する。操作を検出した後は、その操作内容に従って楽音を発音させるための制御データ(発音制御データ)を生成し、それを第2CPU209に転送する。その制御データは、例えばMIDI規格に合わせて生成する。

【0020】スイッチ群208は、モードや音色、テンポといった各種の設定用のスイッチをまとめて表したものである。第1CPU201は、鍵盤102に対して行われた操作を検出すると同様に、キーマトリクス回路207を用いてスイッチ群208をスキャンし、ゲートアレイ206を介してスキャン結果を受け取り、その受け取ったスキャン結果を解析することにより、スイッチ群208を構成する各スイッチの操作状態を検出する。操作状態を検出した後は、その操作状態に従って楽器101の設定を行う。

【0021】上記ゲートアレイ206は、第1CPU201と第2CPU209のインターフェイス等の役割を果たす。第1CPU201が生成した発音制御データは、ゲートアレイ206を介して第2CPU209に転送される。

【0022】上記第2CPU209は、内部に備えたROM(図示せず)に格納されているプログラムを実行することにより、第1CPU201から転送された発音制御データに従って、楽音の発音に関わる制御を実行する。その制御は、RAM210をワーク領域として使用しながら行う。

【0023】発音制御データはMIDIデータの形で第2CPU209に転送される。第2CPU209は、第1CPU201から発音制御データ(MIDIデータ)

6

が転送されると、それを解析して、その発音制御データによって指示される楽音の発音を実現させるための制御データを生成する。生成した制御データはLSI211に送出する。

【0024】LSI211は、RAM211をワーク領域として使用しながら、ROM213に格納されている波形データを用いて、実際に発音させる楽音の波形データを生成する音源LSIである。第2CPU209が送出した制御データに従って、RAM213から波形データを読み出し、その波形データの加算や、指示された音響効果を付加させるための処理を施した後、D/Aコンバータ(DAC)214に出力する。

【0025】D/Aコンバータ214は、LSI211が出力した波形データをアナログのオーディオ信号に変換してアンプ215に出力する。そのアンプ215は、D/Aコンバータ214が出力したオーディオ信号を増幅させてスピーカ216に出力する。アンプ215からオーディオ信号がスピーカ215に入力されることにより、スピーカ216から楽音が放音される。

【0026】上述したように、本実施の形態を搭載した電子楽器101は、楽音を発音させるという本来の機能の他に、画像を表示させる機能を備えている。それらの機能を実現させるための負荷は重いことから、第1及び第2CPU201、209の2つのCPUを用意し、各CPU201、209に、その負荷を分散させる構成となっている。即ち第1CPU201に画像表示を含めた楽器101全体の制御を実行させ、第2CPU209に楽音の発音に関する制御を実行させる構成となっている。

【0027】次に、図2、図3を参照して、本発明に関わる第1CPU201が第2CPU209にデータ転送を行う際の動作について詳細に説明する。上述したように、第1CPU201は発音制御データを第2CPU209に転送することにより、第2CPU209に発音に関する制御を依頼する。第2CPU209が第1CPU201から転送された発音制御データの処理に要する時間は、基本的にデータ量とデータの種類によって決まる。このことから、第1の実施の形態では、発音制御データを構成することができるデータの種類毎に、その処理に要する時間をデータ(発音制御データ処理時間データ)として予めROM202に用意している。

【0028】図3は、ROM202に用意された発音制御データ処理時間データ群の構成図である。発音制御データを構成できるデータとしては、例えばMIDI規格として定められているように、楽音の発音開始を指示するノートオン、発音中の楽音の消音を指示するノートオフ、音色の切り替えを指示するプログラムチェンジ、主に演奏における音響効果の制御用に用いられるコントロールチェンジ等の数多くの種類がある。それらの種類毎に、第2CPU209がその処理に要する時間が16ビ

ットのデータの形でROM202に用意している。図3に示すように、それらのデータがまとまって発音制御データ処理時間データ群を構成している。その第1ワードデータとしてノートオンの処理時間データ、第2ワードデータとしてノートオフのそれ、第3ワードデータとしてプログラムチェンジのそれが格納されている。

【0029】なお、第1の実施の形態では、転送した発音制御データを第2CPU209に確実に処理させるために、各発音制御データ処理時間データを、第2CPU209がそれに対応するデータの処理に必要な最大時間として用いている。しかし、最大時間ではなく、例えば平均的な時間を処理時間データとして用意しても良い。

【0030】図3に示すような発音制御データ処理時間データ群を用意することにより、転送した発音制御データの処理に第2CPU209が必要としている時間を予測することができる。第1CPU201は、発音制御データ処理時間データ群を参照し、発音制御データとして転送するデータの種類、及びデータ量に応じて、それに対応する処理時間データを取得して加算や乗算といった演算処理を行うことにより、第2CPU209が必要とする処理時間を算出する。

【0031】その処理時間を算出することにより、第2CPU209に対して空容量（例えば第2CPU209内部のメモリに割り当てられている領域の空容量である）の問い合わせを行うことなく、その負荷の状態に合わせて発音制御データを転送することができる。例えば、転送した発音制御データに対する処理が終了するのに合わせて発音制御データを第2CPU209に転送することもできるようになる。

【0032】転送された発音制御データの処理が進み、その負荷が小さくなっている状態とは、第2CPU209が空容量を確保する必要がない状態である。このため、空容量を確保する必要がない状態を選んで発音制御データを転送することにより、問い合わせにかかる時間や、第2CPU209が空容量を確保するのにかかる時間を確実に省くことができる。それにより、データ転送に要する時間が短縮されるので、各CPU201、209のスループットを向上させることができ、高い処理効率を得ることができる。

【0033】第1CPU201が実行する主要なサブルーチン処理としては、スイッチ群208を構成している各スイッチの操作状態を検出してそれらの操作状態に応じた設定を行うスイッチ処理、鍵盤102に対して行われた操作を検出して発音制御データを生成する鍵盤処理、自動演奏用のデータを再生するシーケンスデータ再生処理等が挙げられる。これらの各処理は、予め定められた順序に従って繰り返し実行されるようになっており、発音制御データはこれらの処理で生成される。なお、本発明の理解を容易にし、また不要な混乱を避けるために、第1CPU201が実行する主要なサブルーチ

ン処理を、スイッチ処理、鍵盤処理、及びシーケンスデータ再生処理の3つに限定して以降の説明を行うこととする。

【0034】第1の実施の形態では、上記各処理に必要な実行時間をデータとしてROM202に用意している。図4は、ROM202に用意された主要ルーチン実行時間データ群の構成図である。図4に示すように、各データは各々16ビットのデータであり、第1ワードデータとしてスイッチ処理に要する実行時間データ、第2ワードデータとして鍵盤処理に要する実行時間データ、第3ワードデータとしてシーケンスデータ再生処理に要する実行時間データが格納されている。その主要ルーチン実行時間データ群を参照することにより、各処理の実行時間を予測することができるようになっている。

【0035】なお、主要ルーチン実行時間データ群として用意した実行時間データは、全てそれに対応する処理の最大の実行時間としている。しかし、最大の実行時間ではなく、その平均的な実行時間、或いはその最小の実行時間であっても良い。

【0036】第1の実施の形態では、上記各処理に続けて発音制御データを第2CPU209に転送する発音制御データ転送処理を実行するようになっている（図5参照）。各データ転送処理では、それに続いて実行される処理の予測される実行時間と、第2CPU209が転送されている発音制御データに要する予測される残りの処理時間とに基づいてデータ転送を行うようにしている。より具体的には、次に実行する処理で予測される実行時間が、第2CPU209が必要としていると予測される残りの処理時間以下であった場合にはデータの転送を先送りするようにしている。そのようなデータ転送の先送りを行うことにより、第1及び第2CPU201、209に発生するあそび時間を全体として小さくすることができる。このため、従来と比較して、スループットを向上させることができる。

【0037】図5は、第1CPU201が実行する全体処理の動作フローチャートである。次に、図5を参照して、上記第1CPU201の動作について詳細に説明する。図5に示す全体処理は、第1CPU201が、ROM202に格納されているプログラムを読み出して実行することにより実現される処理動作である。

【0038】楽器101の電源がオンされると、始めにステップ501のイニシャル処理が実行され、楽器101が予め定められた状態に設定される。このとき、制御用の各変数の値も初期値に設定される。その変数として、発音制御データ転送処理以外で実行が終了した直前の処理を示すための変数`routine_kind`や、第2CPU209が発音制御データの処理に必要していると予測する処理残り時間を示す値が代入される変数`time_to_wait`がある。それらの変数には各々0が代入される。

9

【0039】ステップ501に続くステップ502では、スイッチ群208をスキャンして各スイッチの操作状態を検出し、それらの操作状態に応じて各種の設定等を行うスイッチ処理を実行する。上記変数 `routine__kind` には、そのスイッチ処理で0が代入される。スイッチ処理が終了すると、ステップ503に移行する。

【0040】ステップ503では、発音制御データを第2CPU209に転送する発音制御データ転送処理を実行する。詳細は後述するように、変数 `time__to__wait` に代入された処理残り時間、次に実行される処理で予測される実行時間に基づいてデータ転送が行われる。

【0041】ステップ503に続くステップ504では、鍵盤102をスキャンしてユーザが行った操作を検出し、その検出した操作に従って発音制御データを生成する鍵盤処理を実行する。この鍵盤処理では、上記変数 `routine__kind` に1を代入する。その鍵盤処理が終了すると、ステップ505で発音制御データ転送処理を実行した後、ステップ506に移行する。

【0042】ステップ506では、ユーザがスイッチ群208の所定のスイッチを介して行った指示に従って、自動演奏用のシーケンスデータを再生するシーケンスデータ再生処理を実行することにより、自動演奏による楽音の発音や、テレビ103の画面上に自動演奏中の曲の楽譜が表示される。変数 `routine__kind` には2が代入される。そのシーケンスデータ再生処理が終了すると、ステップ507で発音制御データ転送処理を実行した後、ステップ502に戻る。

【0043】このように、第1の実施の形態では、スイッチ処理や鍵盤処理といった主要なサブルーチン処理に続けて発音制御データ転送処理を行っている。各処理の実行時間や第2CPU209における処理残り時間は予測した時間であり、多少の誤差がありえる。しかし、図5に示すように、主要なルーチン処理と発音制御データ転送処理を交互に行うようにすると、第1及び第2CPU201、209は全体処理を単位としてではなく、その全体処理を構成するルーチン処理単位で処理を行うことになって、発生する遊び時間が全体的に小さくなる。それにより、上記予測誤差が第1及び第2CPU201、209の遊び時間に反映されるのを抑えることができ、スループットをより向上させることができるようになる。

【0044】次に、上記ステップ503、505、及び507として実行される発音制御データ転送処理について、図6に示すその動作フローチャートを参照して詳細に説明する。

【0045】まず、ステップ601では、前回に発音制御データ転送処理を行ってから現在までの経過時間、例

10

えば今回の発音制御データ転送処理が図5中のステップ503として実行された場合には、ステップ507の発音制御データ転送処理を実行してから現在までの経過時間を算出する。その経過時間は、例えばシステムが備えたフリーランニングタイマ、或いは第1CPU201が備えたタイマ機能を利用して算出する。その算出した経過時間は、変数 `time__interval` に代入されて用いられる。

【0046】ステップ601に続くステップ602では、第2CPU209が必要としていると予測する処理残り時間が経過時間より小さいか否かが判定する。それまでに転送した発音制御データを第2CPU209が全て処理していると予測される場合、その判定はYESとなつてステップ603に移行する。そうでない場合には、その判定はNOとなつてステップ604に移行する。

【0047】ステップ603では、第2CPU209は発音制御データが転送されるのを待っている状態であると予測されることから、処理残り時間を0に設定する。他方のステップ604では、処理残り時間から経過時間を引いた値を処理残り時間に設定する。ステップ603、或いは604が実行されることにより、変数 `time__to__wait` に新たな値が代入された後、ステップ605に移行する。

【0048】ステップ605では、第2CPU209に転送すべきデータが有るか否かが判定する。直前に実行した処理で発音制御データが生成された場合はもとより、それまでに生成した発音制御データのなかで第2CPU209に転送していないデータが有る場合、その判定はYESとなつてステップ606に移行する。そうでない場合には、即ち第2CPU209に転送すべき発音制御データが無い場合には、その判定はNOとなつて一連の処理を終了する。

【0049】ステップ606では、上記ステップ603、或いは604で更新された処理残り時間（変数 `time__to__wait` の値）が0か否かが判定する。発音制御データが転送されるのを第2CPU209が待っていると予測される場合、その判定はYESとなつてステップ609に移行する。そうでない場合には、その判定はNOとなつてステップ607に移行する。

【0050】ステップ607では、変数 `routine__kind` の値が示す他のサブルーチン処理の実行予測時間を算出する。その算出は、上述したように、ROM202に格納されている主要ルーチン実行時間データ群を参照し、変数 `routine__kind` の値に対応する実行時間データを取得することで行われる。それが終了した後、ステップ608に移行する。

【0051】ステップ608では、処理残り時間が、ステップ607で取得した実行予測時間より小さいか否かが判定する。次のサブルーチン処理の実行が終了する前に第2CPU209がその処理を完了することが予測され

11

る場合、その判定はYESとなってステップ609に移行する。そうでない場合には、その判定はNOとなって一連の処理を終了する。

【0052】ステップ609では、第2CPU209に転送すべきデータを実際に転送するデータ転送処理を実行する。そのデータは、第1CPU201からゲートアレイ206を介して第2CPU209に転送される。そのデータ転送が終了すると、ステップ610に移行して、転送したデータに応じた処理残り時間(変数time\_to\_waitの値)の更新を行う。ステップ601で経過時間を算出するための発音制御データ転送処理の実行時刻は、例えばこのステップ610で保持する。そのステップ610の処理を実行した後、一連の処理を終了する。

【0053】このように、発音制御データ転送処理では、第1CPU201が次に実行する処理と第2CPU209が実行している処理の何れが早く終了するかにより、データの転送を行うか否かを決定している。このようなデータ転送を行うことにより、第1CPU201、第2CPU209の各遊び時間の合計が最も小さくなるように、各CPU201、209に処理を実行させることになる。従って、スループットが向上し、高い処理効率を得ることができる。ユーザは、そのスループットの向上を、楽器101に対する操作、例えば鍵盤102を操作したときの反応から認識することができる。

【0054】なお、第1の実施の形態では、発音制御データ転送処理を他のサブルーチン処理と交互に行うようにしているが、必ずしも交互に行うようにしなくとも良い。例えば図5に示す全体処理においては、ステップ502のスイッチ処理は他のステップ504の鍵盤処理等と比較して生成する発音制御データ量が小さいことから、ステップ503の発音制御データ転送処理を行わないようにしても良い。

【0055】また、主要ルーチン実行時間データ群については、第1の実施の形態では各サブルーチン処理毎に1個の実行時間データだけを用意しているが、設定されているモードに応じて実行時間が大きく異なるサブルーチン処理(例えばシーケンスデータ再生処理)も存在する。このことから、モード別に実行時間データを複数用意し、設定されているモードに応じて実行時間データを参照するようにしても良い。

<第2の実施の形態>上記第1の実施の形態では、ROM202に格納されている主要ルーチン実行時間データ群を使用している。しかし、各サブルーチン処理の実際の実行時間は、ユーザの楽器101に対する操作に応じて変化することから、その主要ルーチン実行時間データ群としてROM202に用意した各実行時間データが必ずしも最適であるとは限らない。各実行時間データをより最適な値とすることにより、各CPU201、209のスループットをより向上させることができる。第2の

12

実施の形態は、このことに着目して、各サブルーチン処理のより最適な実行時間を予測できるようにしたものである。

【0056】第2の実施の形態が適用されたシステム構成は、基本的に第1の実施の形態が適用されたそれと同じであり、また、処理動作も基本的に同じである。このため、第1の実施の形態で使用した符号をそのまま使用して、第1の実施の形態から異なる部分のみ説明することにする。

10 【0057】第2の実施の形態では、ROM202に格納された固定値データ群としての主要ルーチン実行データ群(図4参照)に加えて、各サブルーチン処理の実際の実行時間を計測して取得し、それをRAM203に格納して参照するようにしている。図7は、そのようにしてRAM203に格納された主要ルーチン実行時間データ群の構成図である。

20 【0058】図7に示すように、各サブルーチン処理毎に格納領域を用意し、計測した実際の実行時間を対応する格納領域内に格納している。第2の実施の形態では、各格納領域に5個の実行時間データを格納できるようにしている。5個の実行時間データを格納した後は、そのなかで最古の時間データと置き換える。これにより、常に最新の5個の実行時間データを保持するように、実行時間データの更新を行っている。

【0059】第2の実施の形態において、各サブルーチン処理の実際の実行時間は、例えば以下のようにして計測される。図8を参照して説明する。図8は、図5に示す全体処理内でステップ502として実行されるスイッチ処理の動作フローチャートである。

30 【0060】まず、ステップ801では、現在のタイマ(例えばフリーランニングタイマ)値をセーブする。それに続くステップ802では、スイッチ処理におけるメイン処理を実行する。そのステップ802は、第1の実施の形態におけるステップ502と同じ処理である。そのメイン処理が終了すると、ステップ803に移行し、現在のタイマ値からステップ801でセーブしたタイマ値を減算して今回の実行時間データを算出する。その算出した実行時間データは、RAM203のスイッチ処理用の格納領域に格納する。このとき、5個の実行時間データが格納されていなければ空いている場所に、5個格納されていればそのなかで最古の実行時間データが格納されている場所に今回取得した実行時間データを格納する。その実行時間データの格納が終了した後、一連の処理を終了する。

50 【0061】このように、スイッチ処理の始めと終わりの各時点でのタイマ値を参照することで、そのスイッチ処理の実行時間を算出することができる。他の鍵盤処理、及びシーケンスデータ再生処理における実行時間の取得方法も同様であるため、その詳細な説明は省略する。



13

【0062】第1の実施の形態において、次のサブルーチン処理の実行時間の予測は図6に示す発音制御データ転送処理内のステップ607で行われる。第2の実施の形態では、ROM202、及びRAM203にそれぞれ主要ルーチン実行時間データ群を格納していることから、ステップ607では以下のようにして実行予測時間を算出する。

【0063】楽器101の電源がオンされた時点では、RAM203には主要ルーチン実行時間データ群として格納されたデータが存在しない。このため、ROM202に格納されている主要ルーチン実行時間データ群を参照して実行予測時間を算出する。第2の実施の形態では、5個の実行時間データを取得していないサブルーチン処理の実行予測時間はROM202に格納された実行時間データを用いて算出する。

【0064】5個の実行時間データを取得したサブルーチン処理に対しては、ROM202に格納されている実行時間データを参照せずに、その5個の実行時間データに基づいて実行予測時間を算出する。具体的には、5個の実行時間データのなかで最大の実行時間データを抽出し、その実行時間データを用いて実行予測時間を算出する。

【0065】このようにして実行予測時間を算出することにより、サブルーチン処理の実際の実行時間が反映させた発音制御データの転送を行うことができる。そのデータ転送は、過去の履歴に基づいていることから、現実 に即して行うことができる。ステップ608の判定処理による分岐がより高精度に行われるようになる。このため、第1の実施の形態と比較して、各CPU201、209のスループットをより向上させることができる。

【0066】なお、第2の実施の形態では、実行時間データとして取得する個数を5個としているが、それ以外の数としても良い。それらの実行時間データのなかで使用するのは、第2の実施の形態のように最大値ではなく、それらの平均値としても良い。

【0067】また、実行時間は、サブルーチン処理を実行する度に取得するようにしているが、間欠的に実行時間データを取得するようにしても良い。設定されているモードの違いに対応できるように、取得した実行時間をモード別に管理（保持）するようにしても良い。

＜第3の実施の形態＞上記第1及び第2の実施の形態は、共に、サブルーチン処理の実行予測時間と処理残り時間とを比較して、その比較結果から第2CPU209にデータ転送を行うか否かを判定している。しかし、その判定は、各CPU201、209の遊び時間を共に小さくすることに着目したものであり、第2CPU209が転送済みの発音制御データの処理に要する負荷の重さが必ずしも反映されていない。第3の実施の形態は、その第2CPU209の現在の負荷の重さを更に考慮してデータ転送を行うようにしたものである。

14

【0068】第3の実施の形態が適用されたシステム構成は、基本的に第1、或いは第2の実施の形態が適用されたそれと同じであり、また、処理動作も基本的に同じである。このため、第1の実施の形態で使用了符号をそのまま使用して、第1、或いは第2の実施の形態から異なる部分のみ説明することにする。

【0069】第3の実施の形態では、図6に示す発音制御データ転送処理内のステップ609のデータ転送処理だけが第1、或いは第2の実施の形態から異なっている。このため、そのステップ609のデータ転送処理についてのみ、図9に示すその動作フローチャートを参照して詳細に説明する。

【0070】なお、ステップ609のデータ転送処理は、第2CPU209で予測される処理残り時間が0ではなく、且つ第2CPU209に転送すべき発音制御データがある場合に実行される処理である。処理残り時間は0でないことから、その値はステップ604で更新された値である。

【0071】先ず、ステップ901では、ステップ604で更新された処理残り時間が、所定時間より大きいかな否か判定する。その所定時間は、第2CPU209の現在の負荷が重い状態かな否かを判定するための予め定めた定数である。このため、第2CPU209が現在重い状態であると予測される場合、その判定はYESとなってステップ902に移行する。そうでない場合には、その判定はNOとなってステップ903に移行する。

【0072】ステップ902では、第2CPU209の現在の負荷が重い状態となっていることから、第2CPU209に転送すべき発音制御データのなかから、ノートオンデータを削除する。その後、ステップ902に移行する。

【0073】なお、ステップ902でノートオンデータを削除するのは、主に、ノートオンデータの処理に必要な処理時間が比較的に大きいこと、仮にそれを転送しなくとも発音全体に重大な影響を与えない（ここでは、発音がホールドする（鳴りっぱなしになる）といったことがないといった意味である）、という2つの理由からである。ノートオンデータの削除は、例えばRAM203内に確保した第2CPU209に転送すべき発音制御データを格納する領域（バッファ）から、ノートオンデータを削除することで行われる。

【0074】ステップ903は、第2CPU209に転送すべき発音制御データが格納した後に行われる処理である。そのステップ903では、その発音制御データを第2CPU209に実際に転送する処理を実行する。その発音制御データを第2CPU209に転送する処理が、このデータ転送処理におけるメイン処理である。そのデータの転送が終了した後、一連の処理を終了する。

【0075】このように、第3の実施の形態では、第2CPU209の負荷の状態に応じて、転送すべきデータ



15

を選別して第2CPU209に転送する。このため、第2CPU209の負荷が非常に重くなる状況を回避することができ、結果として、各CPU201、209がより効率的に処理を行うことができるようになる。

【0076】なお、第3の実施の形態では、第2CPU209の負荷が重い状態と予測される場合、ノートオンデータを削除するようにしているが、そのデータを削除するのではなく、転送を先送りするようにしても良い。また、削除、或いは先送りするデータとしては、ノートオンデータではなく、発音全体に対する影響が比較的に小さい他の種類のデータとしても良い。データを生成した順序に従って、第2CPU209の負荷の状態に応じて随時転送するようにしても良い。削除、或いは先送りするデータの種類の、第2CPU209の負荷の状態に応じて決めるようにしても良い。

【0077】また、本実施の形態（第1～第3の実施の形態）においては、第2CPU209の負荷の状態を予測だけしているが、その予測誤差を小さくするために、例えばある時間間隔毎といったように、第1CPU201が第2CPU209に対して間欠的にその負荷の状態（未処理のデータ量等）を問い合わせ、予測誤差を修正するようにしても良い。これにより、第2CPU209の状態を常時より高精度に予測することができるようになる。

【0078】本実施の形態のCPU数は2個であるが、当然のことながら、本発明が前提としているCPU数はこれに限定されるものではない。また、本発明は、新製品だけでなく、所定の記憶媒体、或いは通信手段を介して本発明の機能を実現できるプログラムを配布することにより、既存の装置であっても本発明を実現させることができる。電子楽器以外でも本発明を適用することは容易である。

【0079】

【発明の効果】以上説明したように本発明では、転送元（第1CPU）がデータの転送先（第2CPU）にデータを転送する場合、転送先の処理能力、転送済みのデータ量、その種類等から転送先の負荷の状態を予測し、その予測結果に基づいてデータを転送する。このため、転送元は、転送先にその負荷の状態や空容量についての問

16

い合わせを行うことなく、負荷の状態や空容量の状態といった転送先の状態に合わせたデータ転送や、更に割り当てられている処理といった転送元自身の状態も考慮したデータ転送を行うことができる。

【0080】それらのようなデータ転送では、問い合わせに要する時間や、転送先が空容量の確保に要する時間を省くことができる。そのため、データ転送にかかる時間を短縮することができる。そのデータ転送にかかる時間が短縮される結果、転送元、及び転送先のスループットも向上させることができる。

【図面の簡単な説明】

【図1】本実施の形態が適用されたシステムの概観図である。

【図2】本実施の形態が適用されたシステムの構成図である。

【図3】発音制御データ処理時間データ群の構成図である。

【図4】主要ルーチン実行時間データ群の構成図である。

【図5】第1CPUが実行する全体処理の動作フローチャートである。

【図6】発音制御データ転送処理の動作フローチャートである。

【図7】主要ルーチン実行時間データ群の構成図である（第2の実施の形態）。

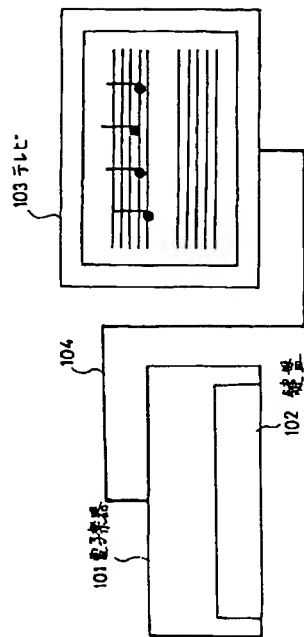
【図8】スイッチ処理の動作フローチャートである（第2の実施の形態）。

【図9】データ転送処理の動作フローチャートである（第3の実施の形態）。

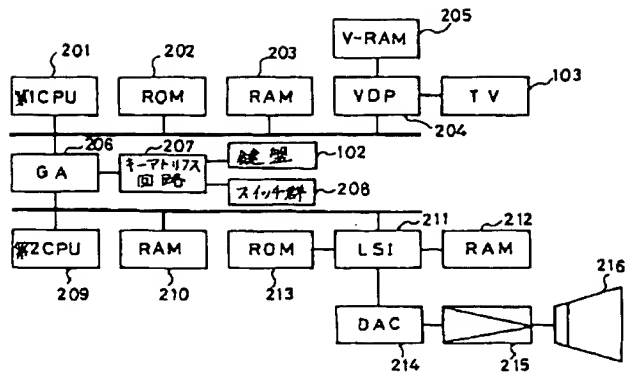
【符号の説明】

101 電子楽器  
102 鍵盤  
103 テレビ  
201 第1CPU  
202 ROM  
204 VDP  
208 第2CPU  
210 LSI

【図 1】



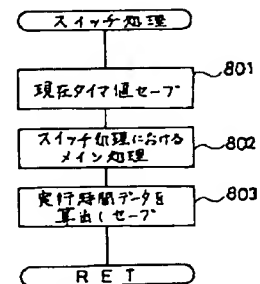
【図 2】



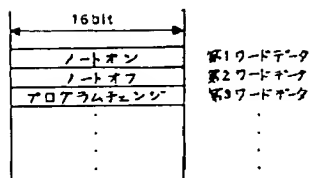
第1の実施の形態が適用されたシステムの構成図

第1の実施の形態が適用されたシステムの概観図

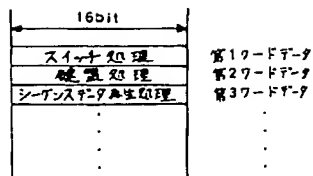
【図 8】



【図 3】



【図 4】

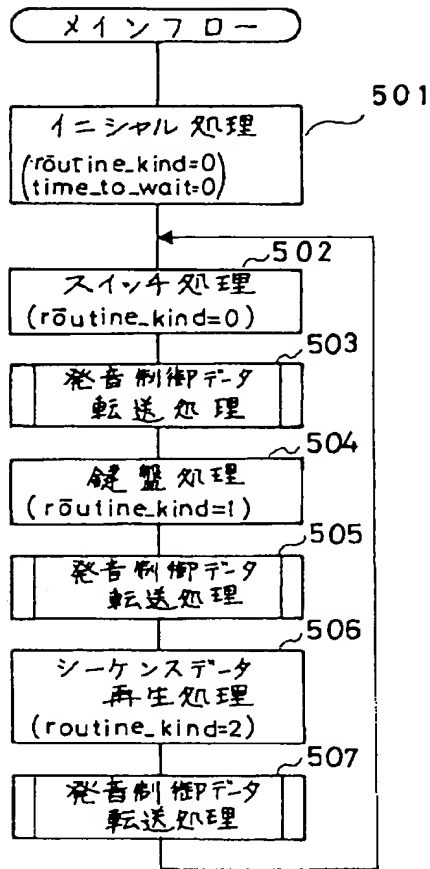


スイッチ処理の動作フローチャート(第2の実施の形態)

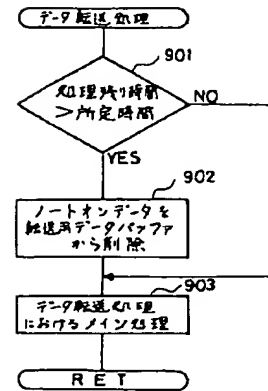
発音制御データ処理時間データ群の構成図

主要ルーチン実行時間データ群の構成図

【図5】



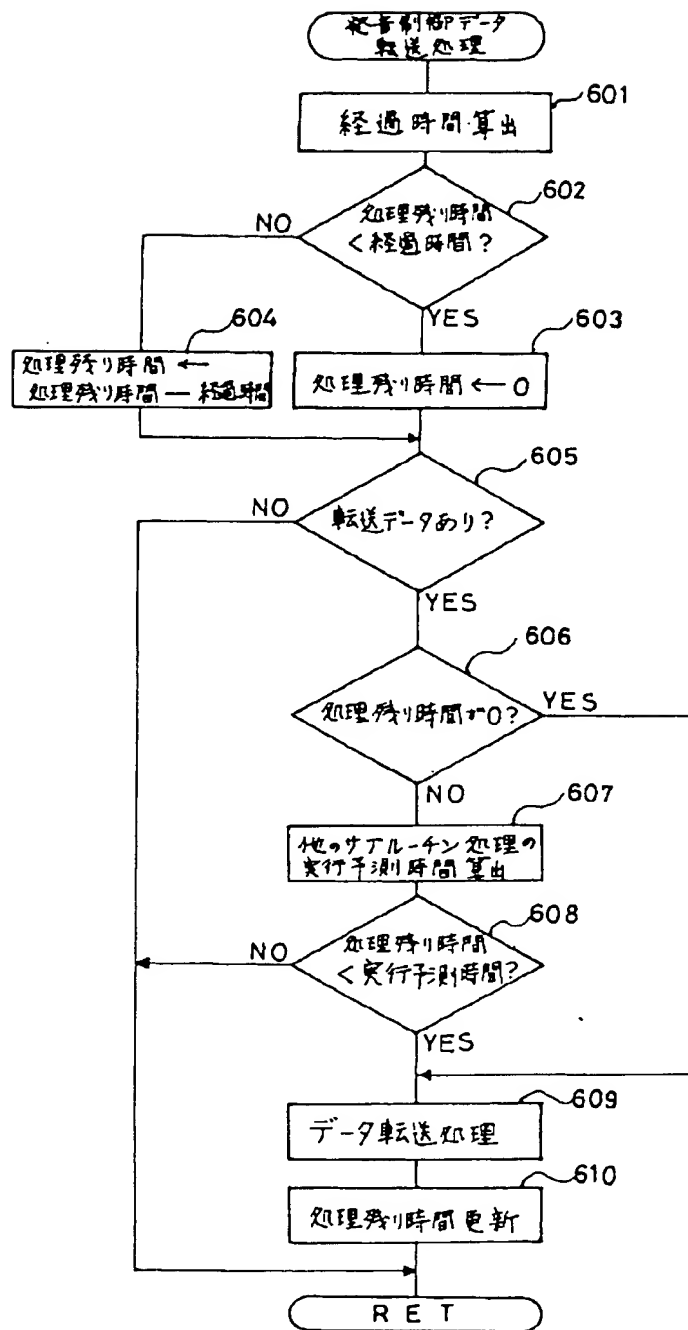
【図9】



データ転送処理の動作フローチャート(第3の実施形態)

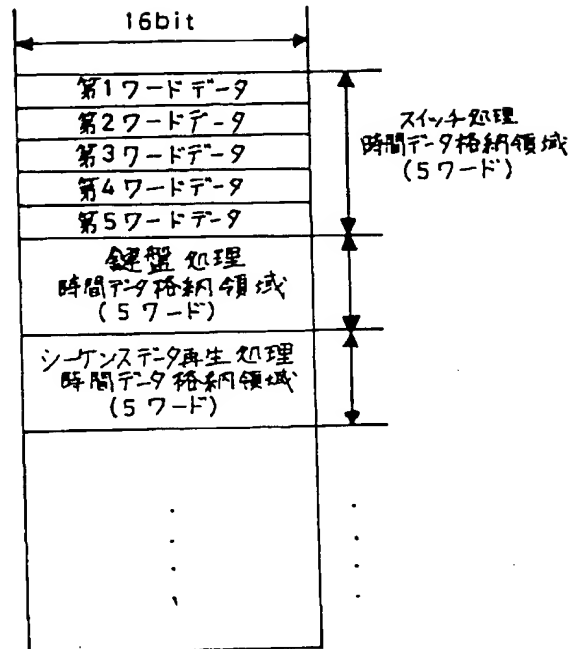
第1CPUが実行する全体処理の動作フローチャート

【図6】



発音制御データ転送処理の動作フローチャート

【図7】



RAMに格納された主要ルーチン実行時間データ群の  
構成図(第2の実施の形態)

**THIS PAGE BLANK (USPTO)**